



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/853,324	05/10/2001	Marappa Kaliappan	450132-03312	5866

20999 7590 12/08/2004

FROMMER LAWRENCE & HAUG  
745 FIFTH AVENUE- 10TH FL.  
NEW YORK, NY 10151

EXAMINER

RUTTEN, JAMES D

ART UNIT PAPER NUMBER

2122

DATE MAILED: 12/08/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

## Best Available Copy

**Office Action Summary**

Application No.

09/853,324

Applicant(s)

KALIAPPAN ET AL.

Examiner

J. Derek Rutten

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 03 August 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 38-82 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 38-82 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 03 August 2004 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### DETAILED ACTION

1. Acknowledgement is made of Applicant's amendment dated 3 August 2004, responding to the 5 May 2004 Office Action provided in the rejection of claims 1-37, wherein claims 1-37 have been canceled, and new claims 38-82 have been added. Claims 38-82 remain pending in the application and have been fully considered by the examiner.
2. Applicant has primarily argued that the claims are not anticipated by Logan because it does not disclose meta-information, nor does it use meta-information to generate test cases. This argument is not persuasive, as will be addressed under the *Response to Arguments* section below.
3. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

***Response to Arguments***

4. Applicant's arguments filed 3 August 2004 have been fully considered but they are not persuasive. Applicant argues on page 15 line 1 of the response that "Logan does not use meta-information". However, review of the cited portion of Logan reveals use of the Java introspection facility (a.k.a. the *Reflection* API) to retrieve such meta-information. See column 7 lines 41-51:

In a preferred embodiment, the test case code generator generates the skeleton test cases by way of the **java.introspection facility from Java's java.lang.reflect package** to test the API calls within a class and provides an exhaustive core API unit test case of a given class. The **introspection facility** of JAVA is provided at run-time to **query a class and identify what methods are in the class**. The introspection facility also determines what arguments a method takes, the data type of each argument, what data types are returned and which exceptions are thrown, if any.

Further support is found in a secondary document that further describes the Java introspection facility (a.k.a. the *Reflection* API) entitled "Take an in-depth look at the Java Reflection API".

See the Summary on page 1:

The process of inspecting components of the program for **meta-information** -- that is, information about the component, as opposed to information about what the component is used for -- is **called introspection**.

The 1.1 version of the Java core classes provides a new API **called the Reflection API**.

Clearly, Logan's use of the Java introspection facility provides meta-information as recited in the claim.

However, In response to applicant's argument in line 2 of page 15 that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., producing a test scenario based on meta-information) are not recited in the

Art Unit: 2122

rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Applicant argues that “the Section 103 rejection” fails as a matter of law since it is only valid under the “obvious to try” standard. As applicant has canceled all original claims, this argument is moot.

### ***Drawings***

5. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the “Boolean foo function” (claims 46 and 61), the “Boundary Value Analysis” with integer and float parameters (claims 47, 48, 62, and 63), and the “Equivalence Partitioning” with a “Boolean having only two test case values” (claims 49 and 64) must be shown or the feature(s) canceled from the claim(s). No new matter should be entered.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as “amended.” If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the

Art Unit: 2122

renumbering of the remaining figures. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

***Claim Rejections - 35 USC § 112***

6. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

7. Claims 46-49 and 61-64 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Claim 46 recites the phrase "regular testing and irregular testing." The originally filed specification does not provide support for these limitations and it is not clear what is meant by these terms. Further, there is no support for a "the **framework** is a Boolean foo function". The specification discloses a framework for testing and provides the example of testing a "Boolean foo function" using a framework on page 21. However, it does not describe wherein the framework is the "Boolean foo function" itself. Claims 47-49 are rejected for being dependent upon a rejected base claim.

Art Unit: 2122

8. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

9. Claims 48, 49, 63, and 64 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 48 and 63 recite: "a parameter of float type having a range between 0 and 1000." However, the numbers "0" and "1000" are not float types, but integers. It is not clear whether the parameter type should instead be referred to as an integer rather than a float, or if the range should be from 0.0 to 1000.0. For the purpose of further examination, the range will be interpreted as "0.0 to 1000.0".

Claims 49 and 64 recite: "a Boolean having only two test case values." It is not clear what "a Boolean" refers to. The word "Boolean" is an adjective that requires a target noun for qualification. Further, it is not clear how such a "Boolean" can have test case values. Is this Boolean a function? Is it a variable? Is it a type of algorithm? Further clarification is required.

### ***Claim Rejections - 35 USC § 102***

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this

Art Unit: 2122

subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

11. Claims 38-41, 43-45, 51-56, 58-60, 66, 72, 74-78, 80, and 81 are rejected under 35

U.S.C. 102(e) as being anticipated by prior art of record, U.S. Patent 6,510,402 to Logan.

In regard to claim 38, Logan discloses:

*A method for testing remote target applications* (column 11 line 61 – column 12 line 57), *said method comprising the steps of:*

*obtaining meta-information of a target application* (column 7 lines 48-53: “The introspection facility of JAVA is provided at run-time to query a class and **identify what methods are in the class**. The introspection facility also determines what arguments a method takes, the data type of each argument, what data types are returned and which exceptions are thrown, if any.”);

*comparing the obtained meta-information with pre-stored meta-information* (column 7 lines 10-13: “In other words, the repository of test suites is managed like a **code repository** with the only difference being that the runtime hook determines if the test suite should be fetched locally or from the remote repository.” Comments: Code repositories operate by comparing differences of stored code.);



*updating the pre-stored meta-information when a discrepancy between the obtained meta-information and the pre-stored meta-information is detected* (column 7 lines 10-13 as cited above discloses repository behavior which updates storage when differences are found.);

*automatically generating test cases based on the obtained meta-information* (column 7 lines 34-39: "...the present invention utilizes a **test case code generator** software program within the ITE clients 206, as described with reference to FIG. 5, to generate a basic core API test skeleton program that systematically exposes all protected methods, constructors and attributes, and makes calls to all methods contained in a subject class.");

*automatically creating test scenarios;* (column 7 lines 3-6: "Further, the present invention utilizes a central repository of production test suites in the ITE server 202. The **test suites are loaded from the repository**, at run time, to perform regression testing") This limitation is interpreted in light of the specification page 10 lines 5-8 which describes automatic generation of test scenarios "using information supplied by the user".

*generating the test cases from the test scenarios* (column 7 lines 51-54: "The test case code generator portion of the ITE client 206 suitably performs **automatic test case generation based on return type** when possible, as well as trapping exceptions.");

*automatically generating test programs using the test scenarios and the test cases*

(column 8 lines 2-6: “Thus, a skeleton **test suite program** is **generated** (step 404) that can then be individualized for specific test cases of the component and integrates into the testing execution framework to automatically **execute each test case** and post test results to the common repository.”);

*building a test image from the test programs* (column 7 lines 7-10: “When test suite code is being developed or revised, it is copied onto an ITE client machine 206 and is then **modified and executed** using the client version, not the server version of the code.” Comment: Modification and execution inherently provides building the image. Without building, the modifications would be unable to execute since they would remain in text form instead of machine executable form.);

*downloading the test image to the target application for testing* (column 7 lines 1-4: “Further, the present invention utilizes a central repository of production test suites in the ITE server 202. The **test suites are loaded from the repository**, at run time, to perform regression testing”);

*automatically testing the target application* (column 8 lines 19-23: “Thus, background tests include those where the expected outcome of the test is known, so that a simple comparison between

Art Unit: 2122

the expected outcome and the actual outcome can be  
**performed in the background** to test for proper  
functioning.”);

*generating reports from test results in a desired format* (column 4 lines 30-32:

“The results of the testing are then stored in the DB server  
214 and **reported to the ITE client 206** by the Web server  
216 (step 226).”);

*providing a framework to define the test scenarios by using the obtained meta-  
information* (column 8 lines 31-33: “FIG. 6 illustrates a diagram of the  
class relationship of the **test execution framework** that run  
on the ITE client 206 in accordance with the present  
invention.”);

*automatically generating different test cases using the test scenarios* (column 7  
lines 41-46: “In a preferred embodiment, the test case code  
generator generates the skeleton test **cases** by way of the  
java.introspection facility from Java's java.lang.reflect  
package to test the API calls within a class and provides  
an exhaustive core API unit test case of a given class.”); and

*generating the test programs in a description language using the test scenarios  
and the test cases* (column 8 lines 2-6: “Thus, a skeleton **test suite  
program is generated** (step 404) that can then be  
individualized for specific test cases of the component and

Art Unit: 2122

integrates into the testing execution framework to automatically **execute each test case** and post test results to the common repository.” Also, column 3 lines 16-22 describes development using the Java Development Kit, wherein the description language is interpreted to be the Java programming language.).

As per claim 39, the above rejection of claim 38 is incorporated. Logan further discloses: *wherein the meta-information details of the target application are obtained using the reflection principle either by the use of reflection object bundled with the target application or by downloading the reflection object to the target application* (column 7 lines 41-51).

As per claim 40, the above rejection of claim 38 is incorporated. Logan further discloses: *wherein the test scenarios, test programs and test image are generated using object serialization in order to improve security of data communication over the network as well as to improve the utilization of resources in the network in order to reduce time for execution* (column 39-42 discloses use of RMI which inherently uses serialization for network transfer).

As per claim 41, the above rejection of claim 38 is incorporated. Logan further discloses: *wherein the said test programs are generated independent of the API or the*

*method for which they are applicable* (column 7 lines 51-54 discloses generation based on return types).

As per claim 43, the above rejection of claim 38 is incorporated. Logan further discloses: *wherein the reports are generated for the specified test scenarios* (column 4 lines 30-32).

As per claim 44, the above rejection of claim 38 is incorporated. Logan further discloses: *wherein the solution is provided to a service station for testing the target application or the said service station is able to use the said automatic test system through a terminal provided at the service station* (column 4 lines 51-58).

As per claim 45, the above rejection of claim 38 is incorporated. Logan further discloses: *wherein a plurality of target applications can be simultaneously tested either at one location or at multiple locations* (FIG. 1 elements 206 depicts testing at multiple locations).

In regard to claim 51, the above rejection of claim 38 is incorporated. Logan further discloses: *wherein the remote target applications are identified with an Internet Protocol (IP) address* (column 3 lines 17-22 describe utilization of JDK 1.1, and in particular the incorporation of Internet standards including IP. The JDK 1.1 inherently

provides location of network elements using an IP address. See “Java™ Platform 1.1 Core API Specification” package java.net, and Class java.net.InetAddress).

In regard to claims 52, Logan discloses: An automatic test system for testing remote target applications (column 12 line 52 – column 14 line 7). All further limitations have been addressed in the above rejection of claim 38

In regard to claim 53, Logan discloses: *wherein the meta-information of the target application is obtained by using a reflection principle by utilizing a reflection object bundled with the target application* (column 3 lines 17-22; also column 7 lines 41-48 describes Java reflection, or introspection. The Reflection API is an inherent feature of Java, and thus is available to every object in every application.).

In regard to claim 54, the above rejection of claim 52 is incorporated. Logan further discloses: *wherein the meta-information of the target application is obtained by using a reflection principle by downloading a reflection object to the target application* (column 7 lines 46-48).

In regard to claims 55, 56, 58-60, and 66, the above rejection of claim 52 is incorporated. All further limitations have been addressed in the above rejection of claims 40, 41, 43-45, and 51, respectively.

In regard to claims 72, 74-78, 80, and 81, the above rejection of claim 52 is incorporated. All further limitations have been addressed in the above rejections of claims 15, 17-21, 23, and 24, respectively.

***Claim Rejections - 35 USC § 103***

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

13. Claims 42, 50, 57 and 65 are rejected under 35 U.S.C. 103(a) as being unpatentable over Logan as applied to claims 38 and 52 above.

In regard to claim 42, the above rejection of claim 38 is incorporated. Logan does not expressly disclose: wherein the execution of the test programs is conducted using the order of execution, the repetition, the requirement for resetting and batch information by user input. It would have been obvious to one of ordinary skill in the art at the time the invention was made to perform execution of the test program as input by the user. One of ordinary skill would have been motivated to provide a system of execution that follows the instructions of the user so that predictable, desired results can be obtained.

In regard to claim 50, the above rejection of claim 38 is incorporated. Logan discloses software testing (column 2 lines 19-22). Logan does not expressly disclose:

*wherein resetting is performed when a user determines that the test programs contain execution errors.* However, Logan discusses conducting repeatable tests in the “Background” section in column 2 lines 4-6. It would have been obvious to one of ordinary skill in the art at the time the invention was made to reset, or repeat a test when errors are found. One of ordinary skill would have been motivated to examine the state of an application prior to the event of an error. The process of testing inherently involves the search for errors. A repeatable test requires that a prior state of execution is reset upon execution of an error.

In regard to claims 57 and 65, the above rejection of claim 52 is incorporated. All further limitations have been addressed in the above rejections of claims 42 and 50, respectively.

14. Claims 67-71 and 73 are rejected under 35 U.S.C. 103(a) as being unpatentable over Logan as applied to claim 52 above, and further in view of prior art of record, “Principles of Object-Oriented Analysis and Design” by Martin (hereinafter “Martin”).

As per claim 67, the above rejection of claim 52 is incorporated. Further, Logan discloses: *configuration* (column 5 line 64 – column 6 line 59), *test design* (column 8 lines 40-42), *test driver* (column 7 line 24-27), *test execution* (column 4 lines 51-58), and *reports* (column 11 lines 16-20), *all connected to data storage through a network* (FIG.



1). Logan does not expressly disclose the organization of functionality into separate specific modules.

However, in an analogous environment, Martin teaches object-oriented (OO) software design which promotes software modules based on object types (page 5 paragraph 2).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to develop Logan's test system using Martin's software modules. One of ordinary skill would have been motivated to represent the functionality of a system in discrete modules that provide a clear conceptual model of a system to promote reliability, flexibility, and ease of change.

As per claim 68, the above rejection of claim 67 is incorporated. Further, Logan discloses: *wherein the configuration module is a software executing on a computing system, which obtains information on test techniques, object details and data type details from the user for defining test cases* (column 5 line 64 – column 6 line 59).

As per claim 69, the above rejection of claim 67 is incorporated. Further, Logan discloses: *wherein the test design module is a software executing on a computing system which provides the test scenario framework to create test scenarios and the information stored in said data storage means* (column 8 lines 40-42).

As per claim 70, the above rejection of claim 67 is incorporated. Further, Logan discloses: *wherein the test driver module is a software executing on computing system, which automatically generates the test cases and then the test programs in a description language using the test scenario provided by the said test design module and the information in said data storage means* (column 7 line 24-27).

As per claim 71, the above rejection of claim 67 is incorporated. Further, Logan discloses: *wherein the test execution module loads an image created by the said image builder module on said target application and monitors and controls the execution of image on said target application* (column 4 lines 51-58).

As per claim 73, the above rejection of claim 67 is incorporated. Further, Logan discloses: *wherein the said report module is a software executing on a computing system for generating reports from the results of the testing on the target application, which are stored in the said data storage means* (column 11 lines 16-20).

15. Claim 79 is rejected under 35 U.S.C. 103(a) as being unpatentable over Logan as applied to claim 1 above, and further in view of “JDBC Guide: Getting Started” by Sun Microsystems (hereinafter referred to as “JDBC”).

As per claim 79, the above rejection of claim 52 is incorporated. Further, Logan discloses: *wherein the said data storage means is a server and is developed in Java*

*making it hardware and software independent using object serialization for communication* (column 3 lines 16-25; also FIG. 1 elements 214, column 3 lines 34-42; also column 4 lines 38-41). Logan does not expressly disclose database specifics.

However, in an analogous environment, JDBC teaches *ODBC / JDBC and is not dependent on any particular database* (Section 1.1 paragraph 2, also Section 1.1.3 paragraph 2).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement Logan's database access using JDBC's database interface. One of ordinary skill would have been motivated to develop a platform independent database application so as to "write it once and run it anywhere".

16. Claim 82 is rejected under 35 U.S.C. 103(a) as being unpatentable over Logan as applied to claim 52 above, and further in view of U.S. Patent 5,919,258 to Kayashima et al. (hereinafter referred to as "Kayashima").

As per claim 82, the above rejection of claim 52 is incorporated. Further, Logan does not expressly disclose the use of a firewall between elements of the communication network.

However, in an analogous environment, Kayashima teaches the use of a firewall to control communication access (Fig. 1; also column 1 lines 34-38).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement Logan's network using a firewall between elements.

Art Unit: 2122

One of ordinary skill would have been motivated to limit the IP address which can access the resources for each service.

### ***Conclusion***

17. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

“An Empirical Analysis of Equivalency Partitioning, Boundary Value Analysis, and Random Testing” by Stuart Reid discloses detailed analysis of well known testing techniques including Boundary Value Analysis (BVA) and Equivalence Partitioning (EP).

“The Java™ Language Specification” by Gosling et al. describes the Java programming language and details the type support that is inherent to the language, including integer, float and Boolean types.


18. Any inquiry concerning this communication or earlier communications from the examiner should be directed to J. Derek Rutten whose telephone number is (571) 272-3703. The examiner can normally be reached on M-F 6:30-3:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2122

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

jdr



**TUAN DAM**  
**SUPERVISORY PATENT EXAMINER**